Computer Science C63 Winter 2024
Scarborough Campus University of Toronto

The Cook-Levin Theorem

Vassos Hadzilacos

Theorem 8.7 (Cook '71, Levin '73) *The satisfiability problem for propositional formulas,* SAT*, is* ***NP****-complete.*

PROOF. It is clear that SAT is in **NP** (the certificate is a truth assignment, which is short, and the verifier checks that the truth assignment satisfies the formula, which can be done in polynomial time). The more interesting part is the proof that SAT in **NP**-hard. Take any decision problem $A \subseteq \Sigma^*$ in **NP**, and let $M_A = (Q, \Sigma, \Gamma, \delta, q_0, h_A, h_R)$ be a nondeterministic Turing machine that decides A in polynomial time, say $p(n)$, where n is the length of the input x. Without loss of generality we can assume that $p(n) \geq n$; we can do this by requiring M_A to read its input before doing anything else. This adds at most n steps to the length of the computation of M_A on x, and therefore does not affect the fact that M_A runs in polynomial time.

Given any $x \in \Sigma^*$ we show how to construct a propositional formula F_x that is satisfiable if and only if $x \in A$; that is, if and only if M_A on input x has an accepting computation $C_0 \vdash C_1 \vdash \cdots \vdash C_\ell$, where $\ell \leq p(|x|)$. The length of F_x will be polynomial in $|x|$, and it will be obvious that it can be constructed from x in polynomial time in $|x|$. Fix any $x \in \Sigma^*$ and let $n = |x|$.

The propositional variables involved in F_x describe the state of affairs in the computation of M_A on input x at each "time" t (i.e., after t steps), $0 \leq t \leq p(n)$. (If $\ell < p(n)$ we will imagine that M_A keeps going until time $p(n)$ without changing its state, head position, or tape contents.) The variables are listed below, along with their intended meaning.

- S_t^q, for $0 \leq t \leq p(n)$ and $q \in Q$: At time t, M_A is in state q.
- H_t^i, for $0 \leq t \leq p(n)$ and $1 \leq i \leq p(n)+1$: At time t, the head of M_A is on cell i. Note that in $p(n)$ steps the rightmost cell that M_A can reach is $p(n)+1$.
- T_t^{ia}, for $0 \leq t \leq p(n)$, $1 \leq i \leq p(n)$, and $a \in \Gamma$: At time t, the cell i of M_A's tape contains symbol a. (Note that all cells to the right of cell $p(n)$ can only contain blanks.)

Thus our formula will have $O\big(p(n)\big) + O\big(p^2(n)\big) + O\big(p^2(n)\big) = O\big(p^2(n)\big)$ variables, i.e., a polynomial number of them. Note that the number of states $|Q|$ and the number of symbols $|\Gamma|$ are constants: these depend on M_A, not on the input x.

The formula F_x is the conjunction of four subformulas, each expressing a requirement for the computation of M_A on input x to be a valid, accepting computation:

(1) Intuitively the subformula F_x^1 states that, at each time t, the variables describe a coherent state of affairs: M_A is in at most one state, its head is in at most one place, and each tape cell contains at most one symbol. (It will follow from this and other subformulas that M_A is actually in exactly one state, the head in exactly one place, and each cell has exactly one symbol.) This is expressed as follows:

$$F_x^1 = \bigwedge_{0 \leq t \leq p(n)} \Bigg(\Big(\bigwedge_{p \neq q \in Q} \underbrace{(\neg S_t^p \vee \neg S_t^q)}_{\text{not in two states}} \Big) \wedge \Big(\bigwedge_{1 \leq i < j \leq p(n)} \underbrace{(\neg H_t^i \vee \neg H_t^j)}_{\text{not in two places}} \Big) \wedge \Big(\bigwedge_{1 \leq i \leq p(n)} \bigwedge_{a \neq b \in \Gamma} \underbrace{(\neg T_t^{ia} \vee \neg T_t^{ib})}_{\text{no two symbols}} \Big) \Bigg).$$

(2) Intuitively the subformula F_x^2 states that the computation of M_A on x starts well: At time 0, M_A is in its initial state q_0, its head is on cell 1, and the tape contains the input x in the first n cells and blanks in cells $n+1..p(n)$. This is expressed as follows: Let $x = a_1 a_2 \dots a_n$, where each $a_i \in \Sigma$.

$$F_x^2 = S_0^{q_0} \wedge \underbrace{\Big(\bigwedge_{1 \le i \le n} T_0^{i a_i} \Big)}_{x \text{ in first } n \text{ cells}} \wedge \underbrace{\Big(\bigwedge_{n+1 \le i \le p(n)} T_0^{i \sqcup} \Big)}_{\text{blanks in rest}}.$$

(3) Intuitively the subformula F_x^3 states that the computation of M_A on x ends well: At time $p(n)$, M_A is in its accept state h_A. This is expressed as follows:

$$F_x^3 = S_{p(n)}^{h_A}.$$

(4) Finally, the subformula F_x^4 is the conjunction of three other formulas, F_x^{4a}, F_x^{4b}, and F_x^{4c}, which intuitively state that the move from time t to time $t+1$ is consistent with the definition of M_A:

- F_x^{4a} states that only the symbol that is under the tape head at time t can change.
- F_x^{4b} states that as long as the current state of the TM is not a halting state (accept or reject), in the next step the state of affairs changes according to the (nondeterministic) transition function of M_A.
- F_x^{4c} states that after reaching a halting state, things don't change.

These are expressed as follows:

$$F_x^{4a} = \bigwedge_{0 \le t < p(n)} \bigwedge_{1 \le i \le p(n)} \bigwedge_{a \in \Gamma} \Big(\neg T_t^{ia} \vee T_{t+1}^{ia} \vee H_t^i \Big).$$

$$F_x^{4b} = \bigwedge_{0 \le t < p(n)} \bigwedge_{q \in Q - \{h_A, h_R\}} \bigwedge_{1 \le i \le p(n)} \bigwedge_{a \in \Gamma} \Bigg(\Big(\neg S_t^q \vee \neg H_t^i \vee \neg T_t^{ia} \Big) \vee \Big(\bigvee_{(p,b,D) \in \delta(q,a)} \big(S_{t+1}^p \wedge H_{t+1}^{i+d} \wedge T_{t+1}^{ib} \big) \Big) \Bigg).$$

where

$$d = \begin{cases} 1, & \text{if } D = R \\ -1, & \text{if } D = L \text{ and } i \neq 1 \\ 0, & \text{otherwise} \end{cases}$$

$$F_x^{4c} = \bigwedge_{0 \le t < p(n)} \bigwedge_{q \in \{h_A, h_R\}} \bigwedge_{1 \le i \le p(n)} \bigwedge_{a \in \Gamma} \Bigg(\Big(\neg S_t^q \vee \neg H_t^i \vee \neg T_t^{ia} \Big) \vee \Big(\big(S_{t+1}^q \wedge H_{t+1}^i \wedge T_{t+1}^{ia} \big) \Big) \Bigg).$$

So, the overall forula F_x is

$$F_x = F_x^1 \wedge F_x^2 \wedge F_x^3 \wedge (F_x^{4a} \wedge F_x^{4b} \wedge F_x^{4c}).$$

Given the semantics of each subformula, F_x asserts that M_A on input x has an accepting computation. In other words,

$$F_x \text{ is satisfiable if and only if } x \in A. \tag{*}$$

IF: If $x \in A$, there is an accepting computation $C_0 \vdash C_1 \vdash \dots \vdash C_\ell$ of M_A on x. From this sequence of configurations we can define truth values for all the variables based on their intended meaning: S_t^q is true if C_t contains the state q and false otherwise, H_t^i is true if in C_t the head is on cell i of the tape and false otherwise, and T_t^{ia} is true of in C_t cell i of the tape contains symbol a and false otherwise). If $\ell < p(n)$ we also define the truth values of the variables corresponding to times t, $\ell < t \leq p(n)$, to be equal to their values at time ℓ. Because $C_0 \vdash C_1 \vdash \cdots \vdash C_\ell$ is an accepting computation of M on x, this truth assignment satisfies all subformulas F_x^1, F_x^2, F_x^3, F_x^{4a}, F_x^{4b}, and F_x^{4c}, and therefore it satisfies their conjunction F_x; therefore, F_x is indeed satisfiable.

ONLY IF: Suppose F_x is satisfiable, and let τ be a truth assignment to the variables that satisfies F_x. From this truth assignment we can define a sequence $C_0, C_1, \ldots, C_{p(n)}$ so that

- C_0 is the initial configuration of M_A on x (this is because τ satisfies F_x^2);
- for each t, $0 \leq t < p(n)$, C_t is a legal configuration of M_A; and either the state of M_A in C_t is not the accept or reject state and $C_t \vdash C_{t+1}$, or the state in C_t is the accept or reject state and $C_t = C_{t+1}$ (this is because τ satisfies F_x^1, F_x^{4a}, F_x^{4b}, and F_x^{4c}); and
- $C_p(n)$ is a configuration in the accept state (this is because τ satisfies F_x^3.

Therefore for some ℓ, $0 \leq \ell \leq p(n)$, $C_0 \vdash C_1 \vdash \ldots \vdash C_\ell$ is an accepting computation of M_A on x, which means that $x \in A$. This completes the proof of $(*)$.

Now let us calculate the length of F_x, measured as the number of variable occurrences. Recalling that $|Q|$ and $|\Gamma|$ are constants we see that the lengths of F_x^1, F_x^2, F_x^3, F_x^{4a}, F_x^{4b}, and F_x^{4c} are, respectively, $O(p^3(n))$, $O(p(n))$, $O(1)$, $O(p^2(n))$, $O(p^2(n))$, and $O(p^2(n))$. (For F_x^{4b} note that the maximum number of choices due to the nondeterminism of M_A is $(|Q|-2) \cdot |\Gamma|$, which is constant.) Therefore the size of F_x is polynomial in n (the length of the input x), and obviously can be constructed from x in polynomial time.

So there is a polynomial time mapping reduction from any decision problem in **NP** to SAT. $\square$

The formula F_x in the proof of the Cook-Levin theorem is almost in conjunctive normal form (CNF). Only the subformulas F_x^{4b} and F_x^{4c} are not. We will now show that we can put these subformulas in CNF without sacrificing the polynomial size of the resulting formula, by using the distributive law (of disjunctions over conjunctions). For example, consider F_x^{4b}. To simplify the notation, note that this formula is a conjunction of formulas of the form

$$\phi = (\ell_1 \vee \ell_2 \vee \ell_3) \vee \underbrace{\Big(\bigvee_{i=1}^{k} (\ell_{i1} \wedge \ell_{i2} \wedge \ell_{i3})\Big)}_{\phi'}$$

where ℓ_1, ℓ_2, ℓ_3 and the ℓ_{ij}s are literals (i.e., variables or negated variables), and k is the number of choices of M_A's nondeterministic transition function, a constant. We can put ϕ' in CNF by applying the distributive law, resulting in the following equivalent formula:

$$\phi'' = \bigwedge_{\pi \in \{1,2,3\}^{\{1,2,\ldots,k\}}} \big(\ell_{1\pi(1)} \vee \ell_{2\pi(2)} \vee \ldots \vee \ell_{k\pi(k)}\big).$$

(If X and Y are sets, Y^X denotes the set of all functions from X to Y. Thus a function $\pi \in \{1,2,3\}^{\{1,2,\ldots,k\}}$ maps each $i = 1, 2, \ldots, k$ to 1, 2, or 3. Intuitively, π selects one of the three literals of each clause in ϕ'.) By replacing ϕ' by ϕ'' in ϕ and applying the distributive law once more (now of disjuctions over conjunctions) we get that ϕ is equivalent to the following formula

$$\psi = \bigwedge_{\pi \in \{1,2,3\}^{\{1,2,\ldots,k\}}} \big(\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_{1\pi(1)} \vee \ell_{2\pi(2)} \vee \ldots \vee \ell_{k\pi(k)}\big).$$

which is in CNF. The length of ϕ, measured as the number of variable occurrences, is $3k+3$, whereas that of ψ is $3^k(k+3)$. But recall that k is a constant and therefore the size of F_x, with F_x^{4b} replaced by an equivalent CNF formula as above, remains polynomial in the size of the input x. F_x^{4c} is similar but simpler, since in this case $k=1$. Therefore we have:

Corollary 8.8 *The satisfiability problem for CNF formulas,* CNF-SAT*, is **NP**-complete.*

It turns out that the satisfiability problem remains **NP**-complete even if we restrict it to CNF formulas where each clause has at most three literals. Such formulas are called 3-CNF and the corresponding satisfiability problem is called 3SAT.

Theorem 9.1 3 SAT *is **NP**-complete.*

PROOF. 3SAT is in **NP** because it is a special case of SAT, which is in **NP**. To prove that CNF-SAT $\leq_m^p$ 3SAT, we will show how to replace each clause C of a CNF formula F by a 3-CNF formula C' that is satisfiable if and only if C is.

If C has at most three literals, we just take $C' = C$. Otherwise, let

$$C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ldots \vee \ell_k$$

for some $k > 3$, where $\ell_1, \ldots, \ell_k$ are literals. Let z_1 be a new variable that does not appear anywhere else in F and consider the formula

$$C_1 = (\ell_1 \vee \ell_2 \vee z_1) \wedge (\neg z_1 \vee \ell_3 \vee \ldots \vee \ell_k).$$

Intuitively C_1 says that (at least) one of ℓ_1, ℓ_2, z_1 is true, and that, furthermore, if z_1 is true then (at least) one of $\ell_3, \ldots, \ell_k$ is true. Thus, C is satisfiable if and only if C_1 is satisfiable. Note that whereas C has k literals, C_1 has two clauses, one with three literals and one with $k-1$ literals. If $k-1>3$ we apply the same idea recursively to the second clause, obtaining another formula C_2 that has two clauses with three literals and one with $k-2$ literals and is satisfiable if and only if C is. We repeat this until we obtain a conjunction of clauses each of which has at most three variables, i.e., a 3-CNF formula. This is the formula C' by which we replace C:

$$C' = (\ell_1 \vee \ell_2 \vee z_1) \wedge (\neg z_1 \vee \ell_3 \vee z_2) \wedge (\neg z_2 \vee \ell_4 \vee z_3) \wedge \ldots \wedge (\neg z_{k-3} \vee \ell_{k-1} \vee \ell_k).$$

Note that C' has at most 3 times as many literals as C, so F' is linear in the size of F. So in polynomial time we can construct a 3-CNF formula F' that is satisfiable if and only if the CNF formula F is. $\square$

What if we further restrict CNF formulas so that each clause has at most two literals? The satisfiability problem for such formulas, called 2SAT, turns out to be solvable in polynomial time!